NAME
rarc – ra client resource file.

SYNOPSIS
rarc

COPYRIGHT
Copyright (c) 2000-2011 QoSient. All rights reserved.

DESCRIPTION
Ra* clients will open this file if its in the users $HOME directory, or in the $ARGUSHOME directory, and parse it to set common configuration options. All of these values will be overriden by options set on the command line, or in the file specified using the '-F conffile' option.

Values can be quoted to make string denotation easier, however, the parser does not require that string values be quoted. To support this, the parse will remove " (double quote) characters from input strings, so do not use this character in strings themselves.

Values specified as "" will be treated as a NULL string, and the parser will ignore the variable setting.

RA_ARGUS_SERVER
All ra* clients can attach to a remote server, and collect argus data in real time. This variable can be a name or a dot notation IP address. Optionally you can specify a port number using a ':' and then providing the port number desired.

RA_ARGUS_SERVER=localhost:561

RA_CISCONETFLOW_PORT
All ra* clients can read Cisco Netflow records directly from Cisco routers. Specifying this value will alert the ra* client to open a UDP based socket listening for Cisco Netflow data on the port number specified.

RA_CISCONETFLOW_PORT=

RA_OUTPUT_FILE
All ra* clients can support writing output as Argus Records into a file or stdout. Stdout is specified as '-'.

RA_OUTPUT_FILE="filename"

RA_TIMERANGE
All ra* clients can support input filtering on a time range. The format is:
timeSpecification[-timeSpecification]

where the format of a timeSpecification can be:
[[[yy/]mm/]dd.]hh[:mm[:ss]]
[yy/]mm/dd

RA_TIMERANGE="55/12/04.00:00:01-55/12/04.23:59:59"
RA_TIMERANGE="12/04-12/05"

RA_RUN_TIME
All ra* clients can support running for a number of seconds, while attached to a remote source of argus data. This is a type of polling. The default is zero (0), which means run indefinately.

RA_RUN_TIME=0

RA_PRINT_LABELS

Most ra* clients are designed to print argus records out in ASCII, with each client supporting its own output formats. For ra() like clients, this variable will generate column headers as labels. The number is the number of lines between repeated header labeling. Setting this value to zero (0) will cause the labels to be printed once. If you don't want labels, comment this line out, delete it or set the value to -1.

RA_PRINT_LABELS=0

RA_FIELD_DELIMITER

Most ra* clients are designed to print argus records out in ASCII, with each client supporting its own output formats. For ra() like clients, this variable can overide the default field delimiter, which are variable spans of space (' '), to be any character. The most common are expected to be '' for tabs, and ',' for comma separated fields.

RA_FIELD_DELIMITER=','

RA_PRINT_NAMES

For **ra(1)** like clients, this variable will control the translation of various numbers to names, such as address hostnames, port service names and/or protocol names. There can be a huge performance impact with name lookup, so the default is to not resolve hostnames.

RA_PRINT_NAMES=port

Other valid options are **none** to print no names, **proto** to translate the protocol names, **port** to translate port names, and **all** to translate all the fields. An invalid option will default to **port**, silently.

RA_PRINT_RESPONSE_DATA

For ra() like clients, this variable will include the response data that is provided by Argus. This is protocol and state specific.

RA_PRINT_RESPONSE_DATA=no

RA_PRINT_UNIX_TIME

For ra() like clients, this variable will force the timestamp to be in Unix time format, which is an integer representing the number of elapsed seconds since the epoch.

RA_PRINT_UNIX_TIME=no

RA_TIME_FORMAT

For ra() like clients, the format that is used to print timestamps, is based on the strftime() library call, with an extension to print fractions of a sec using "%f". The default is "%T.%f". You can overide this default time format by setting this variable. This string must conform to the format specified in strftime(). Malformed strings can generate interesting output, so be aware with this one, and don't forget the '.' when doing fractions of a second.

RA_TIME_FORMAT="%T.%f"

RA_TZ

The timezone used for timestamps is specified by the tzset() library routines, and is normally specified by factors such as the TZ environment variable found on most machines. You can override the TZ environment variable by specifying a time zone using this variable. The format of this string must conform to the format specified by tzset(3).

RA_TZ="EST5EDT4,M3.2.0/02,M11.1.0/02"
RA_TZ="PST8PDT"

RA_USEC_PRECISION

For ra() like clients, this variable is used to override the time format of the timestamp. This variable specifies the number of decimal places that will be printed as the fractional part of the time. Argus collects usec precision, and so a maximum value of 6 is supported. To not print the fractional part, specify the value zero (0).

RA_USEC_PRECISION=6

RA_USERDATA_ENCODE

Argus can capture user data. When printing out the user data contents, using tools such as raxml(), the type of encoding can be specified here. Supported values are "Ascii", or "Encode64".

RA_USERDATA_ENCODE=Ascii

RA_DEBUG_LEVEL

If compiled to support this option, ra* clients are capable of generating a lot of use [full | less | whatever] debug information. The default value is zero (0).

RA_DEBUG_LEVEL=0

RA_FILTER

You can provide a filter expression here, if you like. It should be limited to 2K in length. The default is to not filter. See ra(1) for the format of the filter expression.

RA_FILTER=""

SASL SUPPPORT

When argus is compiled with SASL support, ra* clients may be required to authenticate to the argus server before the argus will accept the connection. This variable will allow one to set the user and authorization id's, if needed. Although not the best practice, you can provide a password through the RA_AUTH_PASS variable. If you do this, you should protect the contents of this file. The format for this variable is:

RA_USER_AUTH="user_id/authorization_id"
RA_AUTH_PASS="password"

The clients can specify a part of the negotiation of the security policy that argus uses. This is controlled through the use of a minimum and maximum allowable protection strength values. Set these variable to control this policy.

RA_MIN_SSF=0

RA_MAX_SSF=128

If compiled to support this option, ra* clients are capable of generating a lot of use [full | less | whatever] debug information. The default value is zero (0).

RA_DEBUG_LEVEL=0

Some ra style clients use a non-blocking method to connect to remote data sources, so the user many need to control how long to wait if a remote source doesn't respond. This variable sets the number of seconds to wait. This number should be set to a reasonable value ($5 <$ value < 60). The default value is 10 seconds.

RA_CONNECT_TIME=10

Some ra* clients have an interval based function. Ratop, as an example, can refresh the screen at a fixed interval. This variable can be set using the RA_UPDATE_INTERVAL variable, which is a float in seconds. 0.5 seconds is the default.

RA_UPDATE_INTERVAL=0.5

All ra* clients have the ability to print country codes for the IP addresses that are in a flow record. Country codes are generated from the ARIN delegated address space files. Specify the location of your DELE-GATED_IP file here.

No Commandline equivalent

RA_DELEGATED_IP="/usr/local/argus/delegated-ipv4-latest"

SEE ALSO
    ra(1)